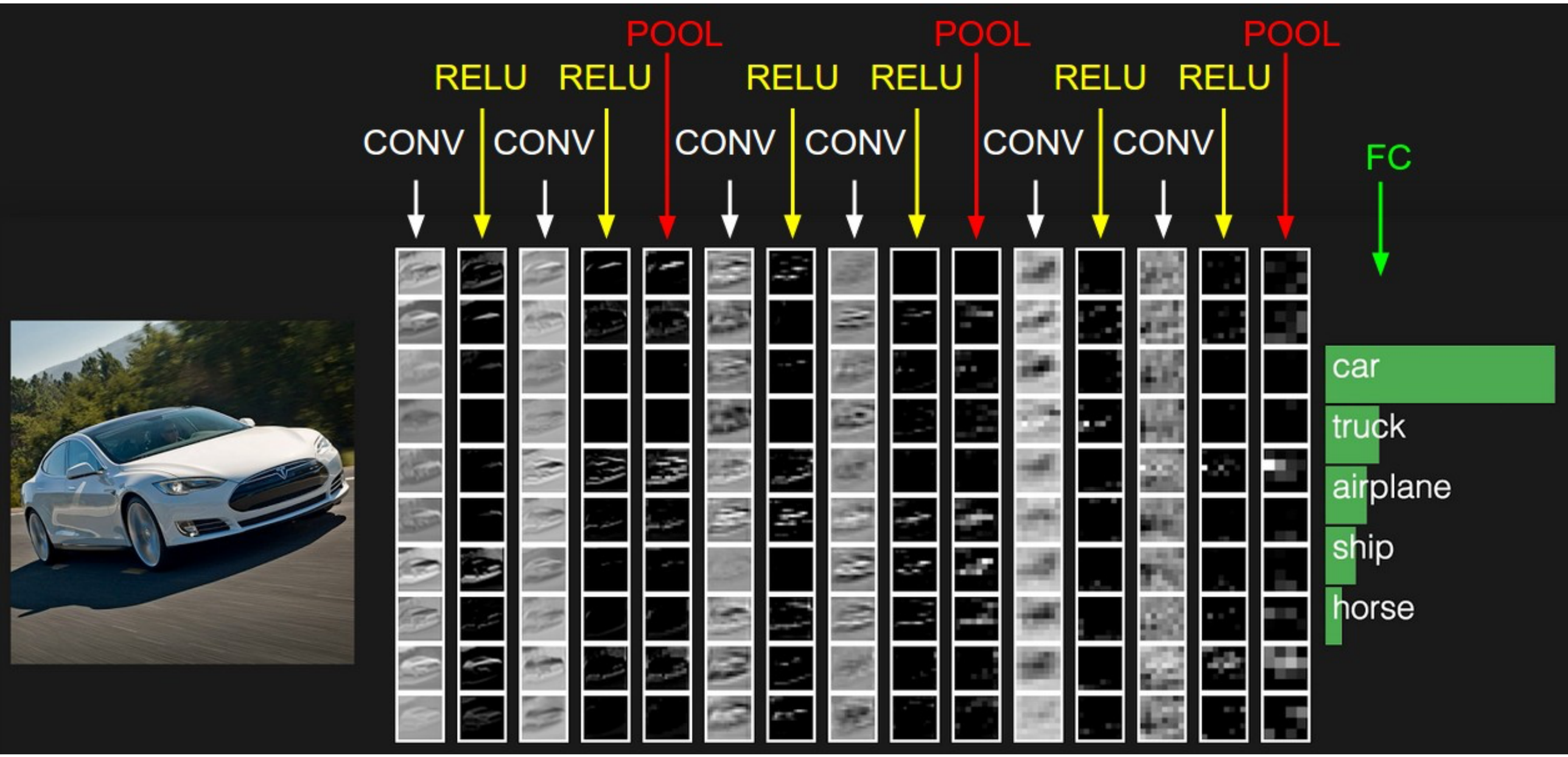
Image Classification Networks: classical architectures and common design patterns

Kate Farrahi

Vision, Learning and Control
University of Southampton

Motivation: Image Classification



ImageNet Dataset

- ImageNet is a massive dataset of annotated photographs for computer vision research

- Collected by researchers at Stanford, led by Prof. Fei-Fei Li

- +14M images, +21K classes, +1M images with bounding box annotations

- Image annotations by humans using crowdsourcing (e.g. Amazon's Mechanical Turk)

- The database of annotations of third-party image URLs is freely available, though the images are not owned by ImageNet.

History of ImageNet

- Fei-Fei Li began working on the idea for ImageNet in 2006. At that time, most AI research focused on models and algorithms.

- In 2007, Li met with Princeton professor Christiane Fellbaum, one of the creators of WordNet. As a result of this meeting, Li went on to build ImageNet starting from the word-database of WordNet and using many of its features.

- WordNet is a lexical database of semantic relations between words in more than 200 languages. Began in the mid 80s by George Miller.

- They presented the database for the first time as a poster at the 2009 CVPR Conference in Florida (paper).

https://en.wikipedia.org/wiki/ImageNet

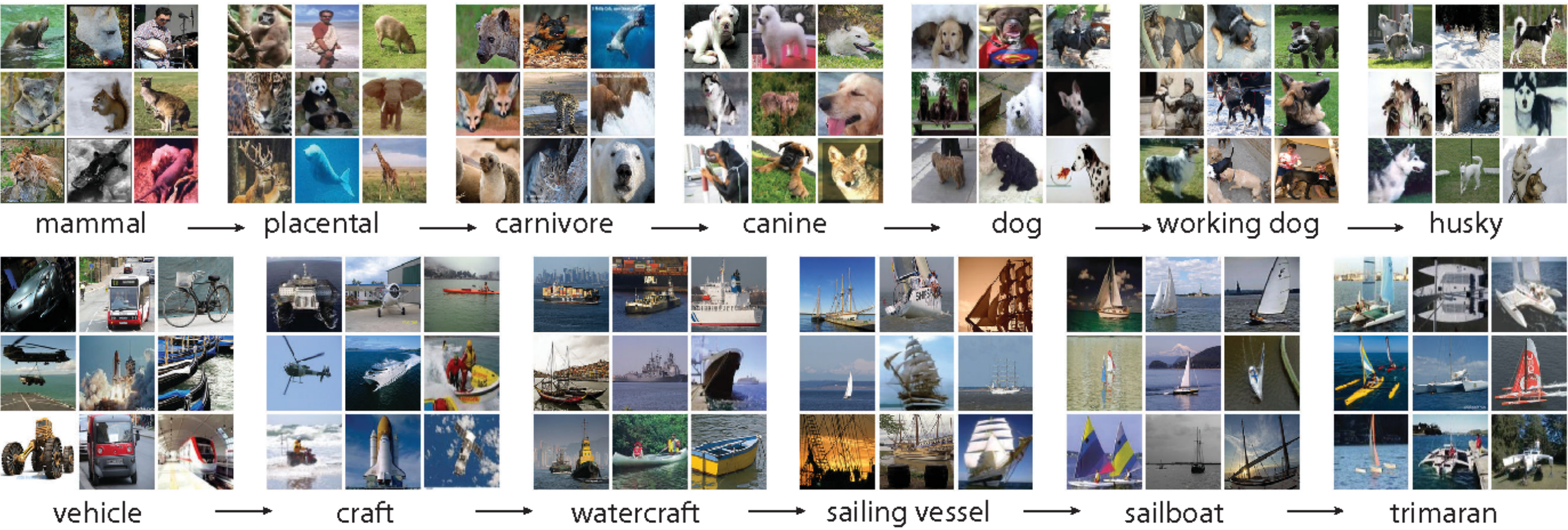


Image taken from the ImageNet paper CVPR 2009.

The ImageNet Challenge

- ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- Annual challenge on a subset of ImageNet

- Designed to foster the development and benchmarking of state-of-the-art algorithms

- The challenge has led to milestone model architectures and techniques that are more widely used in deep learning

Download

Download ImageNet Data

The most highly-used subset of ImageNet is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012-2017 image classification and localization dataset. This dataset spans 1000 object classes and contains 1,281,167 training images, 50,000 validation images and 100,000 test images. This subset is available on Kaggle.

For access to the full ImageNet dataset and other commonly used subsets, please login or request access. In doing so, you will need to agree to our terms of access.

ILSVRC

- The annual competition was held between 2010 and 2017 using subsets of the ImageNet dataset

- Typically, the training dataset comprised of 1 million images, with 50,000 for a validation dataset and 150,000 for a test set (available to download here)

- The general challenge tasks for most years (Image classification, Single-object localization, Object detection)

- Publication: Int J Comput Vis paper link, TED talk

Image classification

Steel drum

Ground truth

Steel drum
Folding chair
Loudspeaker

Accuracy: 1

Scale
T-shirt
Steel drum
Drumstick
Mud turtle

Accuracy: 1

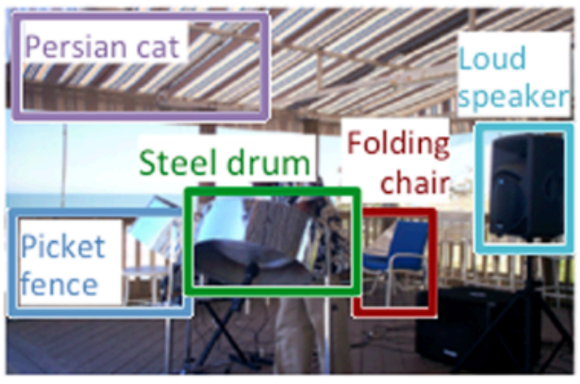
Scale
T-shirt
Giant panda
Drumstick
Mud turtle

Accuracy: 0

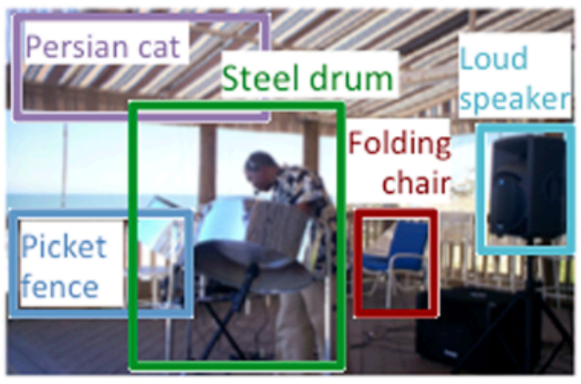
Single-object localization

Steel drum

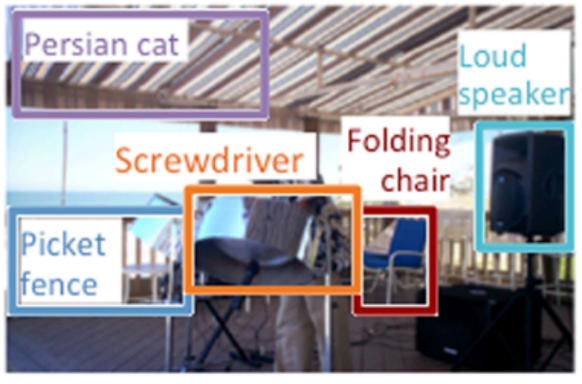
Ground truth



Accuracy: 1

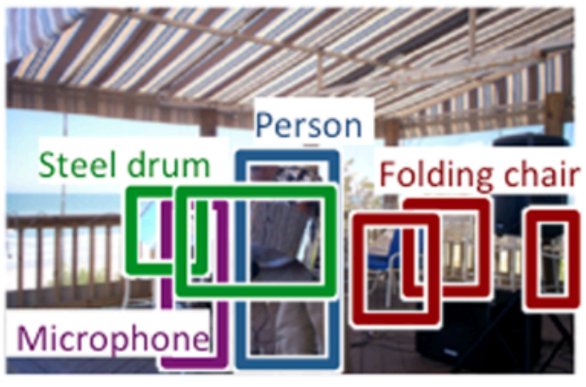


Accuracy: 0

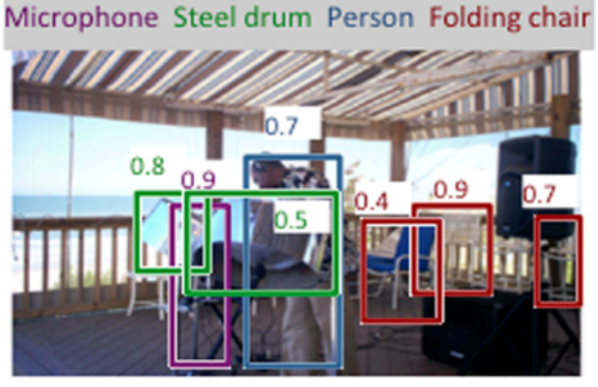


Accuracy: 0

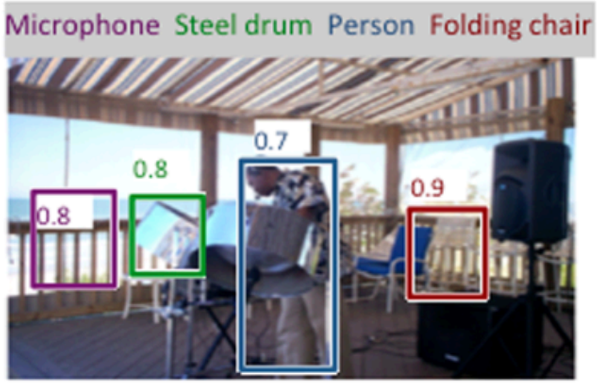
Object detection



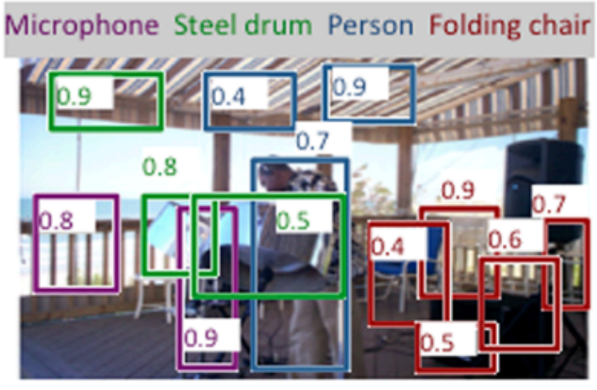
Ground truth



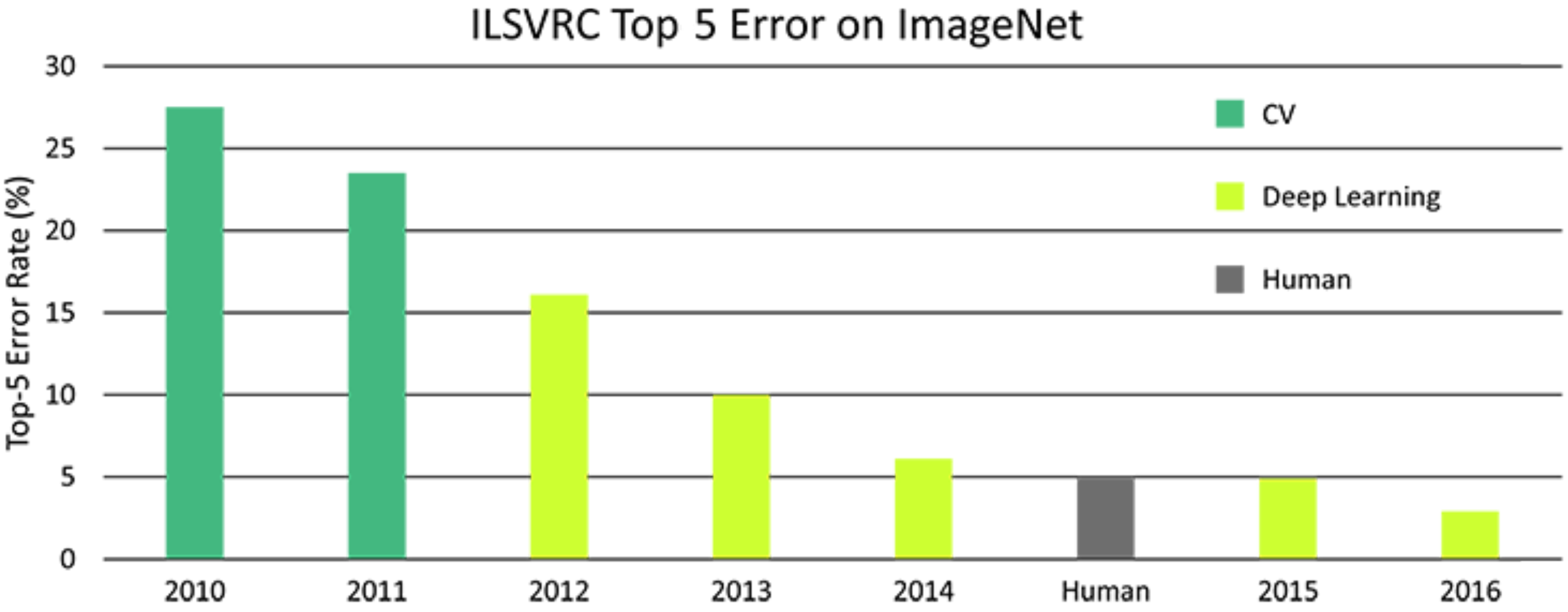
AP: **1.0** **1.0** **1.0** **1.0**

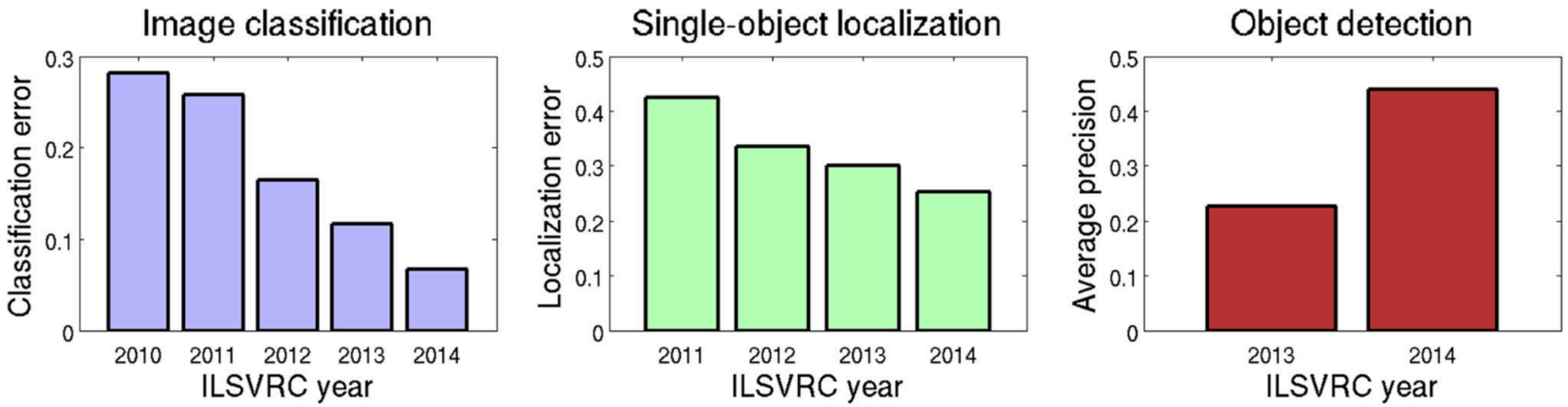


AP: **0.0** **0.5** **1.0** **0.3**



AP: **1.0** **0.7** **0.5** **0.9**


ILSVRC Top 5 Error on ImageNet
Top-5 Error Rate (%)
30
25
20
15
10
5
0
2010
2011
2012
2013
2014
Human
2015
2016
CV
Deep Learning
Human



Taken from paper ImageNet IJCV 2015 (link)

Image Classification on ImageNet

Leaderboard Dataset

View Top 5 Accuracy by Date for All models

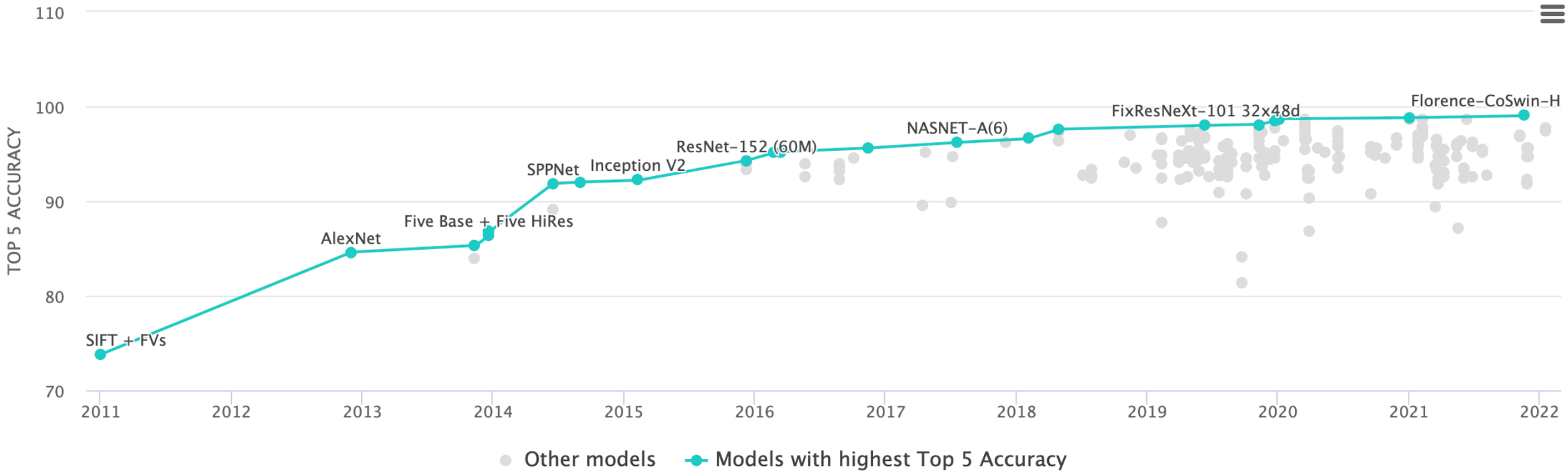


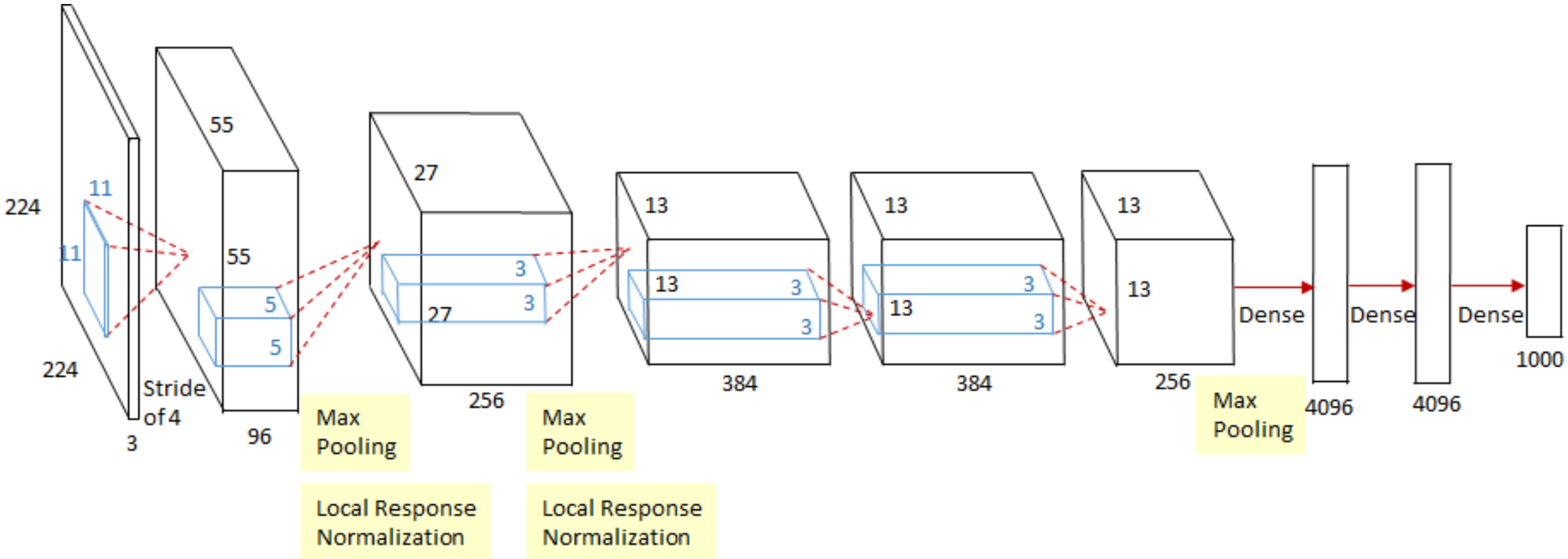
Image taken from here

Classic Architectures

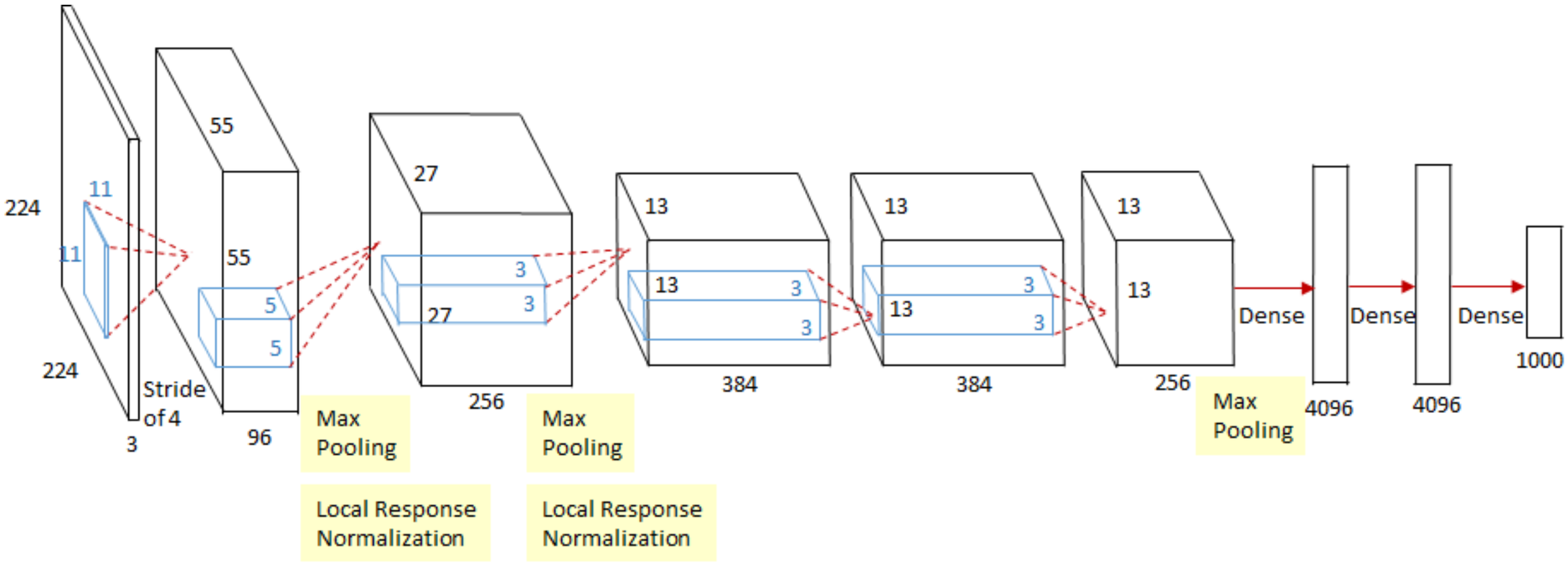
AlexNet

ImageNet Classification with Deep Convolutional Neural Networks. https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

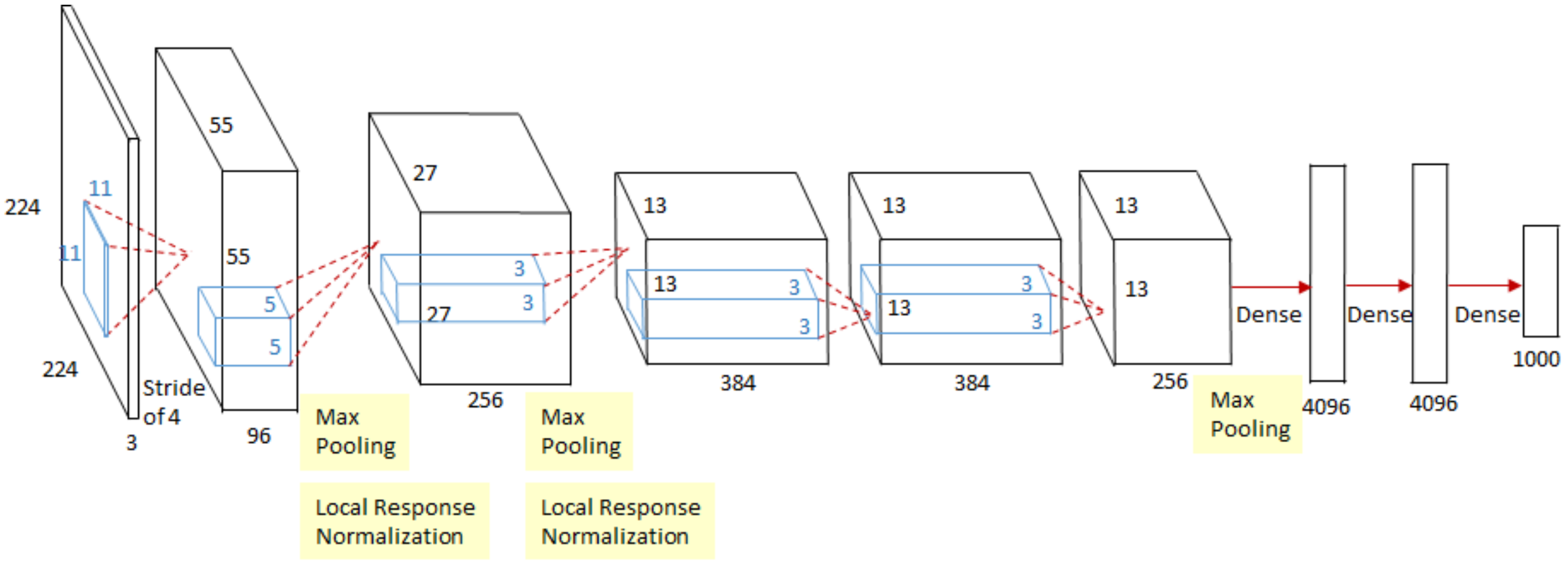
CaffeNet


224
11
11
224
3
Stride
of 4
55
55
96
5
5
Max
Pooling
Local Response
Normalization
27
27
256
3
3
Max
Pooling
Local Response
Normalization
13
13
384
3
3
13
13
384
3
3
13
13
256
Max
Pooling
Dense
4096
Dense
4096
Dense
1000

CaffeNet - How many parameters?



CaffeNet - Effective Receptive Field Sizes



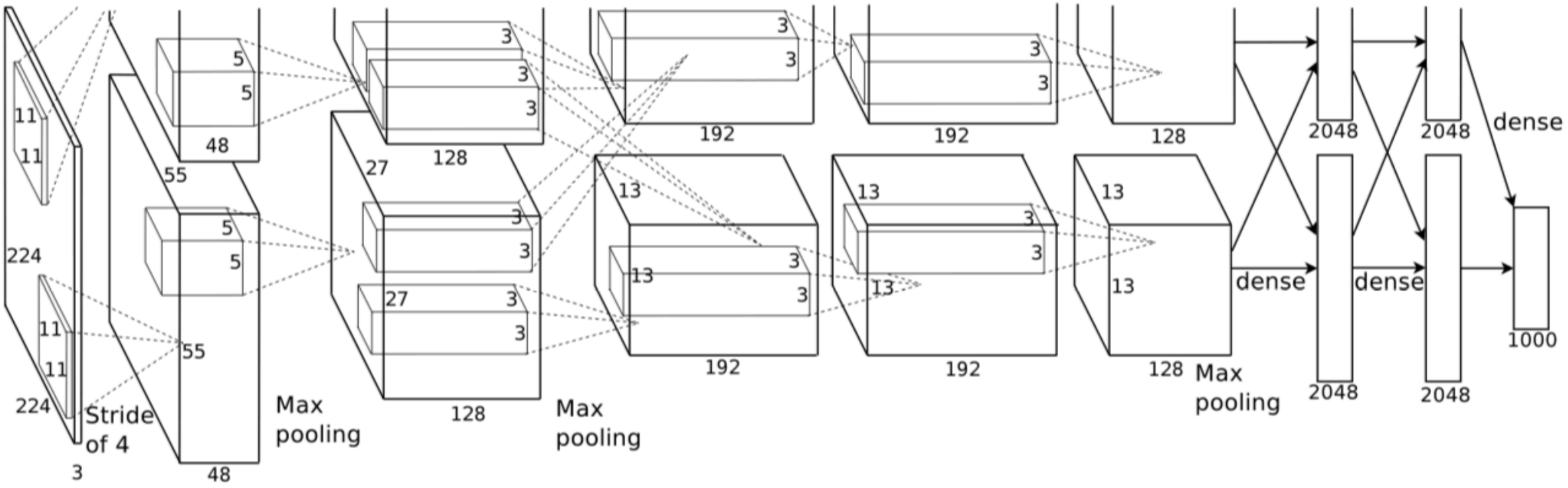
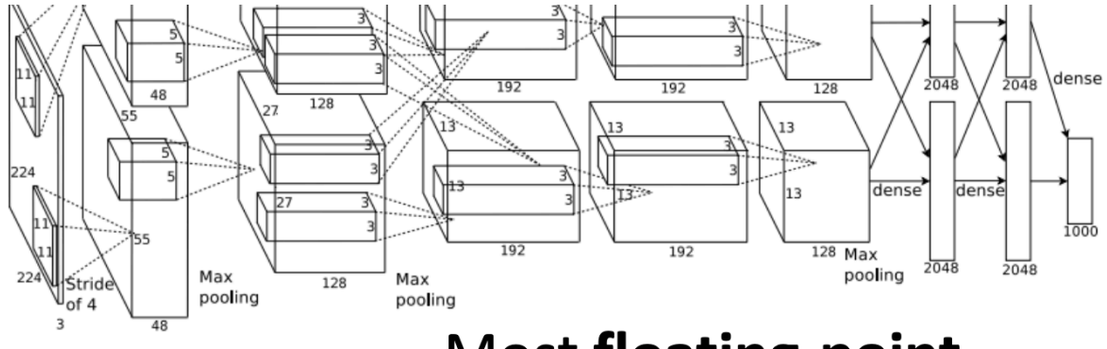


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

AlexNet

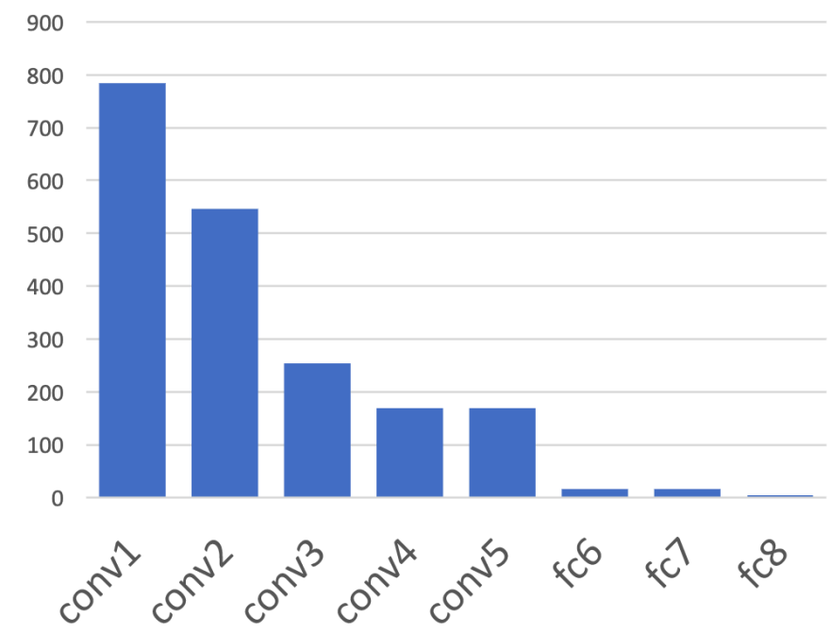


Most of the **memory usage** is in the early convolution layers

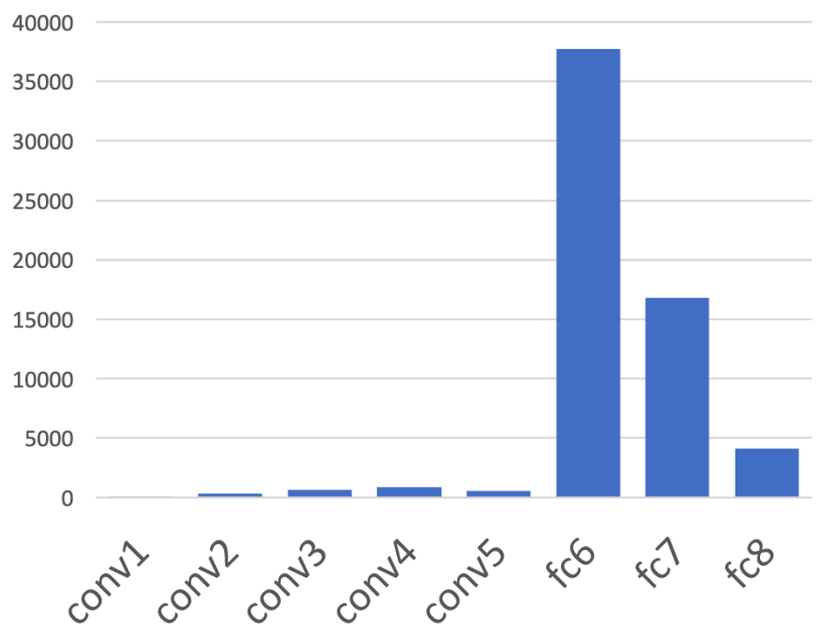
Nearly all **parameters** are in the fully-connected layers

Most **floating-point ops** occur in the convolution layers

Memory (KB)



Params (K)



MFLOP

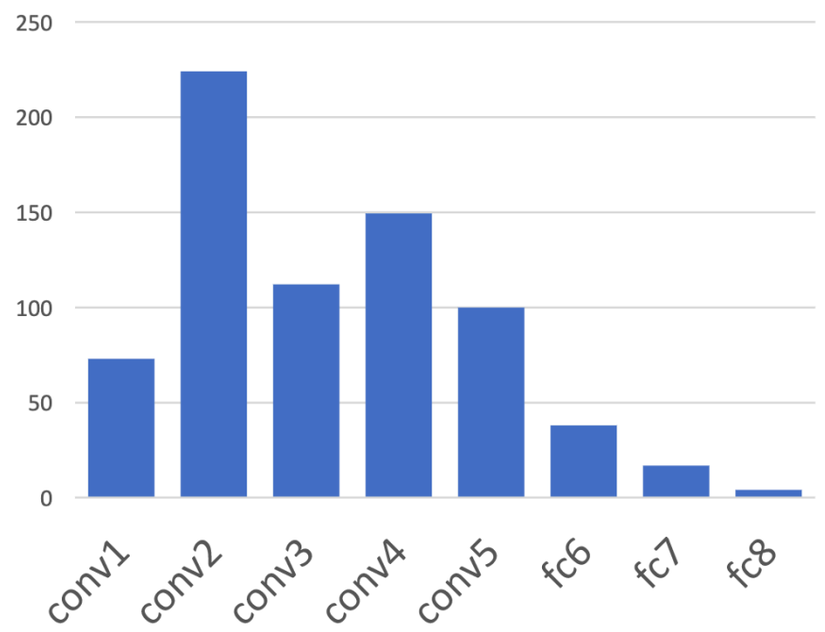
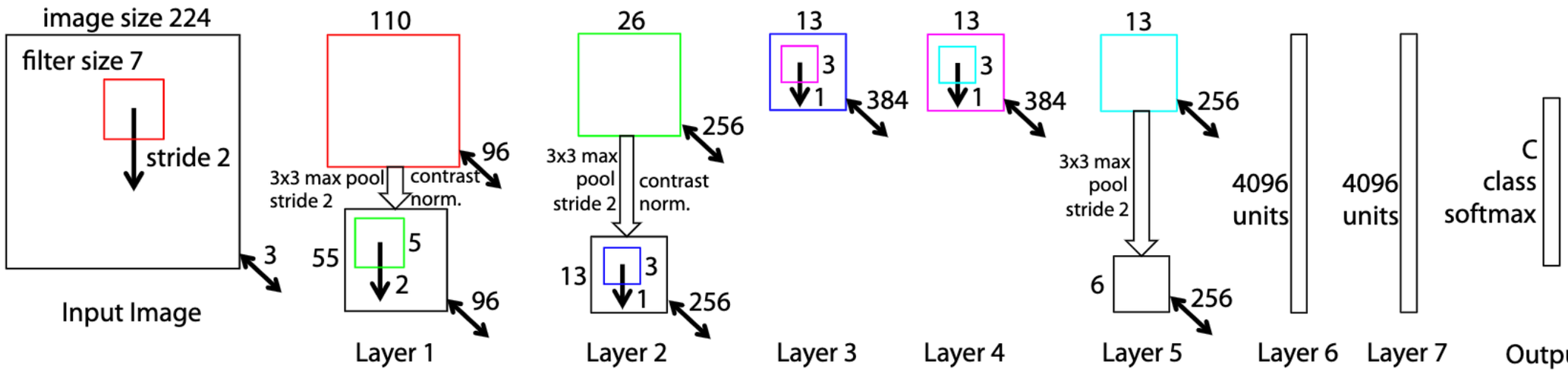


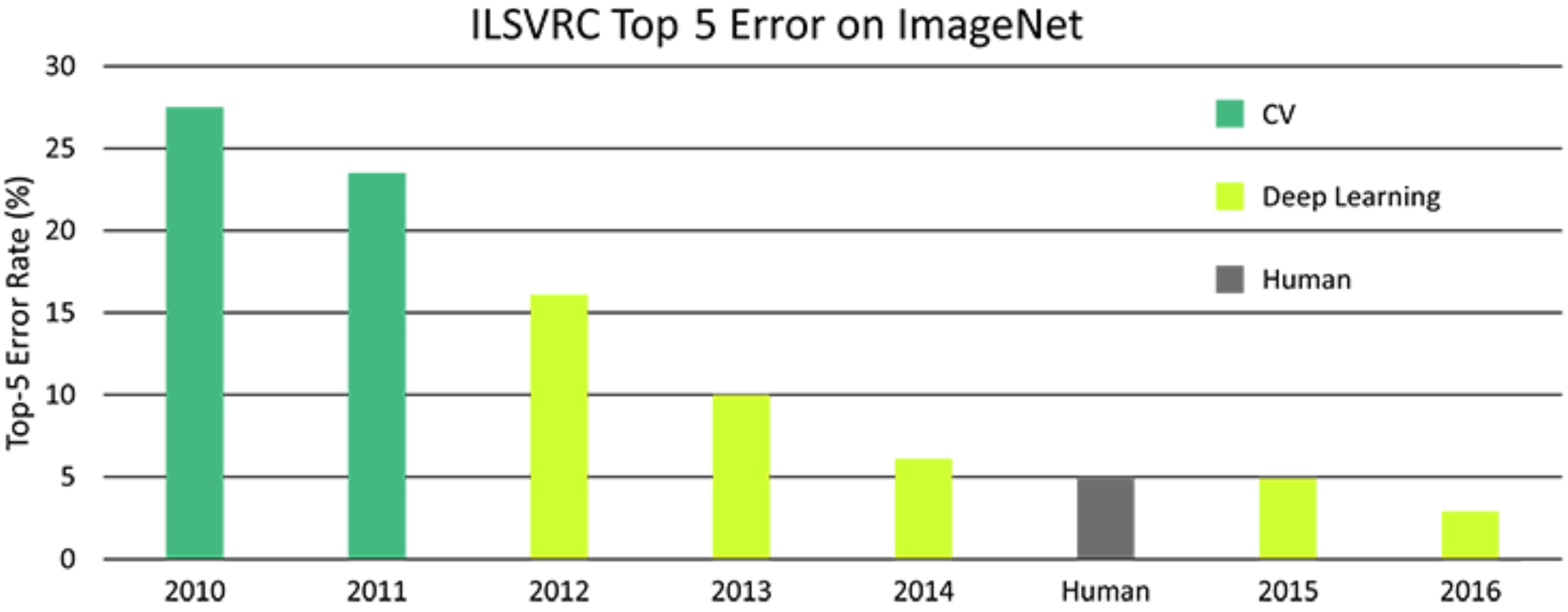
Image from J. Johnson

ZFNet



- ILSVRC 2013 winner
- Essentially "a bigger AlexNet"
- ImageNet top 5 error: 16.4% —> 11.7%
- Applied trial and error to improve the convolutional layer parameters

Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.


ILSVRC Top 5 Error on ImageNet
Top-5 Error Rate (%)
30
25
20
15
10
5
0
2010
2011
2012
2013
2014
Human
2015
2016
CV
Deep Learning
Human

The All CNN

Striving for Simplicity: The All Convolutional Net. https://arxiv.org/pdf/1412.6806.pdf

The All CNN

- A simpler approach using all CNNs
- Aiming for efficiency by removing the FC layers at the end that generate most of the parameters
- Tested on CIFAR10 and CIFAR100 datasets (small scale)
- The novelty is in how they reduce the number of parameters at the end of the network, which is used in some of the later models.
- The network works quite well, and doesn't have many weights.
- Top 1 error of 41.2% on ImageNet in comparison to AlexNet performing 40.7% with 6 times fewer parameters (<10M parameters)

Model

A	B	C
Input 32×32 RGB image		
5×5 conv. 96 ReLU	5×5 conv. 96 ReLU 1×1 conv. 96 ReLU	3×3 conv. 96 ReLU 3×3 conv. 96 ReLU
3×3 max-pooling stride 2		
5×5 conv. 192 ReLU	5×5 conv. 192 ReLU 1×1 conv. 192 ReLU	3×3 conv. 192 ReLU 3×3 conv. 192 ReLU
3×3 max-pooling stride 2		
3×3 conv. 192 ReLU		
1×1 conv. 192 ReLU		
1×1 conv. 10 ReLU		
global averaging over 6×6 spatial dimensions		
10 or 100-way softmax		

The VGG Networks

Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org/pdf/1409.1556.pdf

VGG

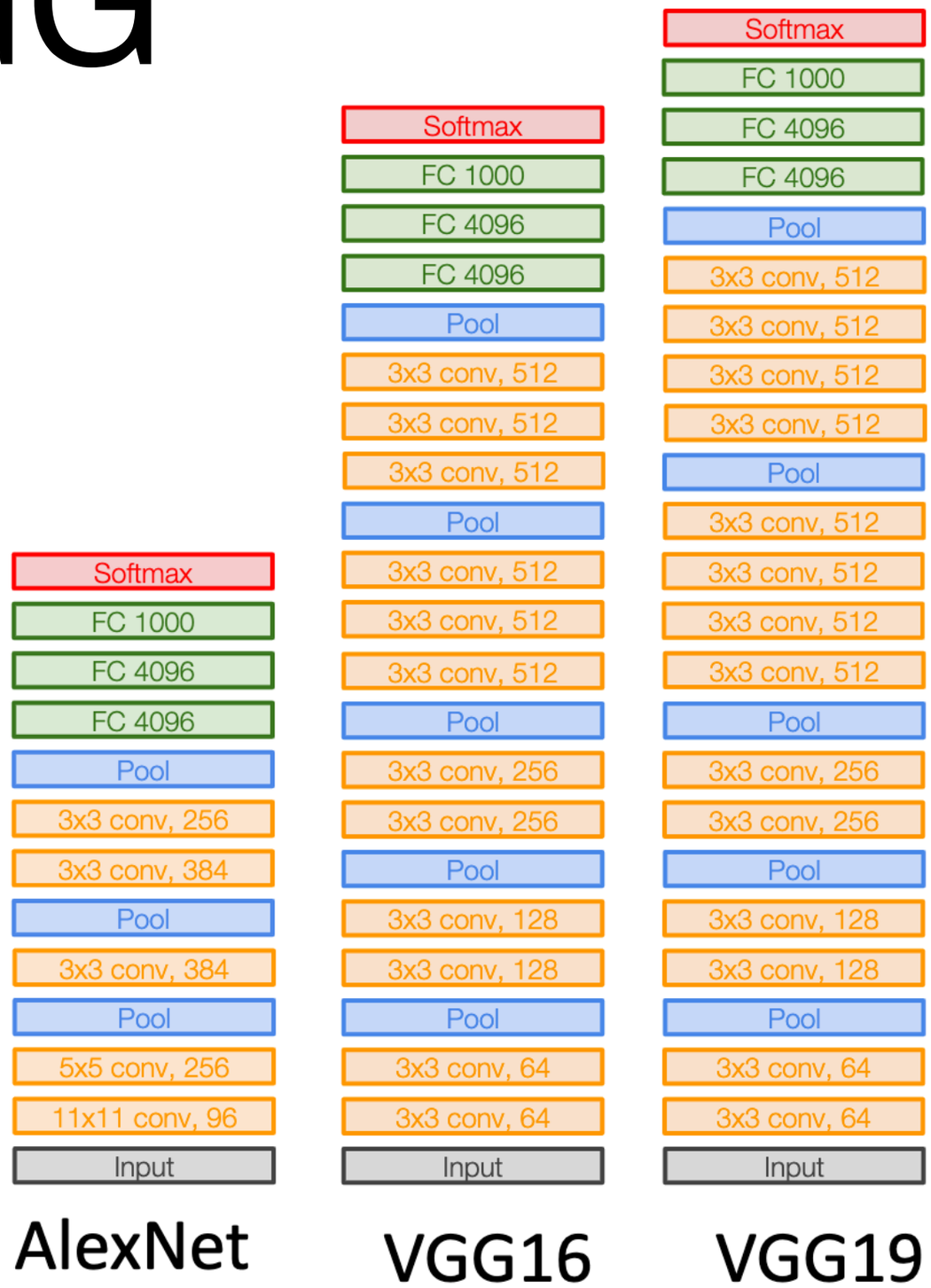
- All conv 3x3 S1 same padding
- All maxpool 2x2 S1
- After Pool, double #channels

- VGG16 — 16 layers w learnable parameters - 5 convolution stages

- VGG19 — 19 layers 5 convolution stages (4 conv in stages 4 and 5)

- 5x5xC x C (25C^2 params)
- 2 x 3x3xC x C (18C^2 params)

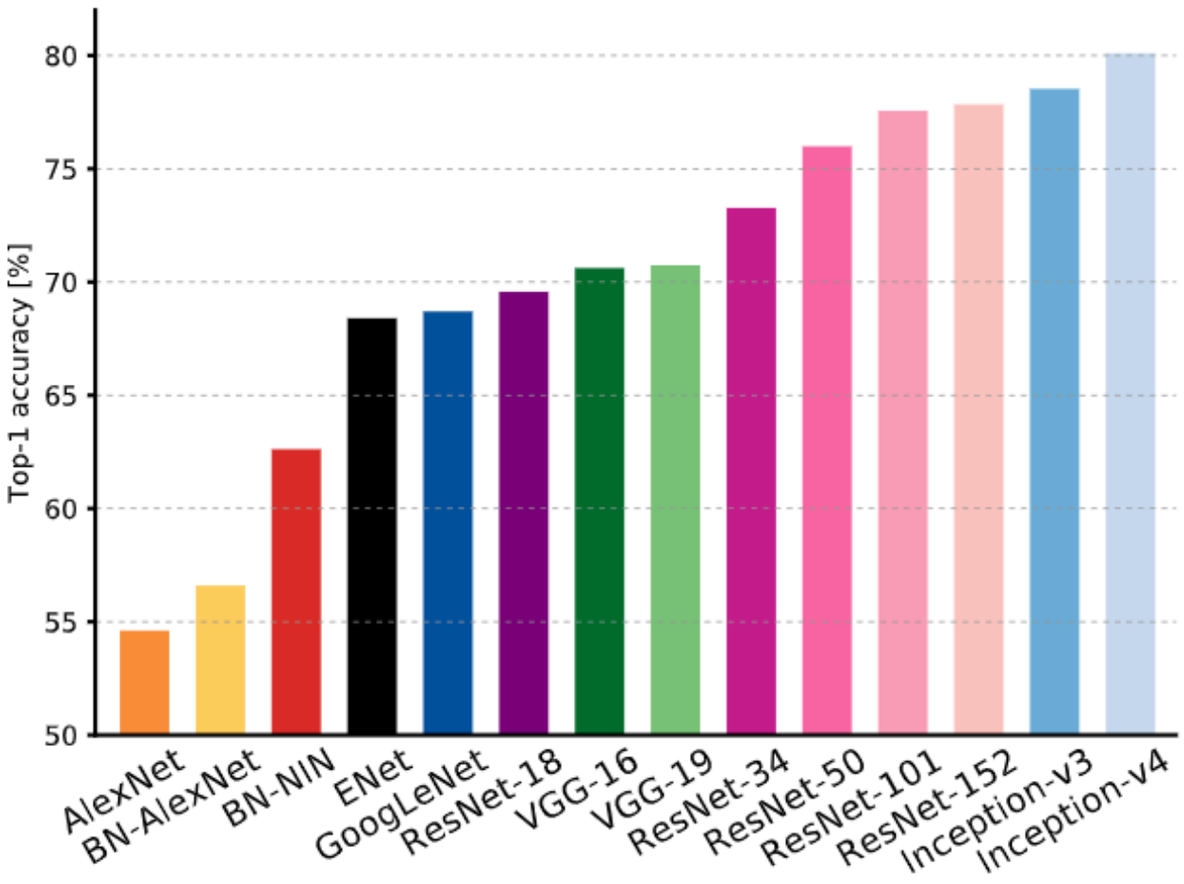
- Two 3x3 conv have same receptive field as a single 5x5 conv, but has fewer parameters and takes less computation!

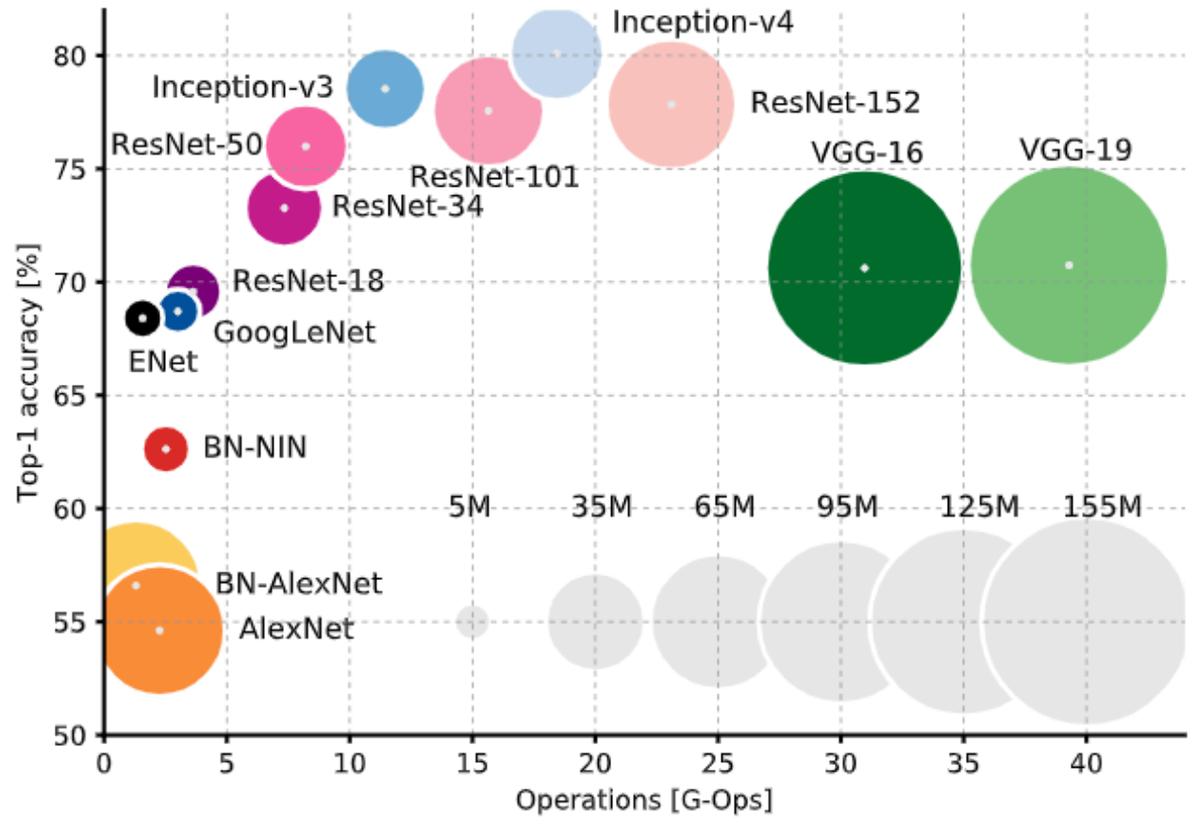


ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

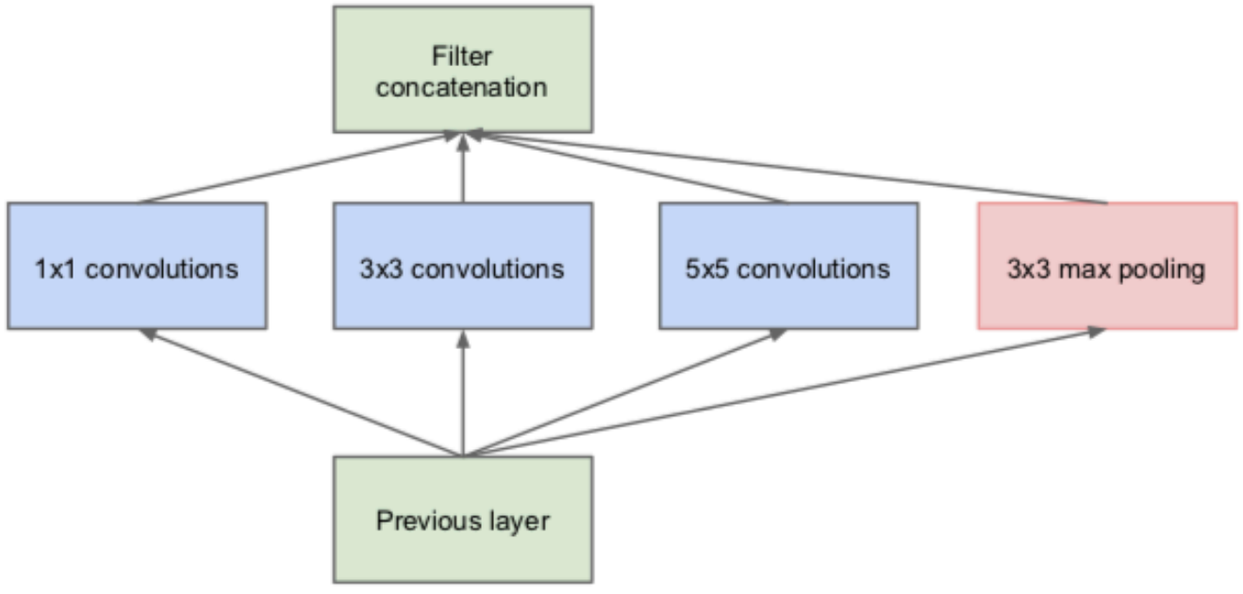




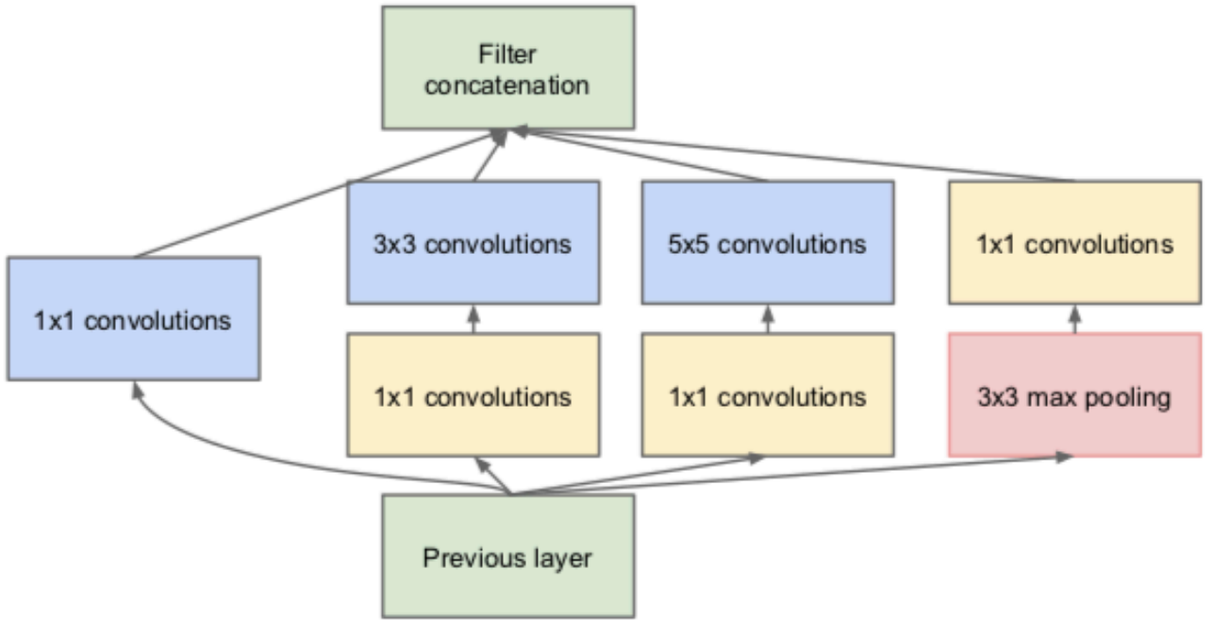
An Analysis of Deep Neural Network Models for Practical Applications. https://arxiv.org/pdf/1605.07678.pdf

GoogLeNet and the Inception Module

Going Deeper with Convolutions. https://arxiv.org/pdf/1409.4842



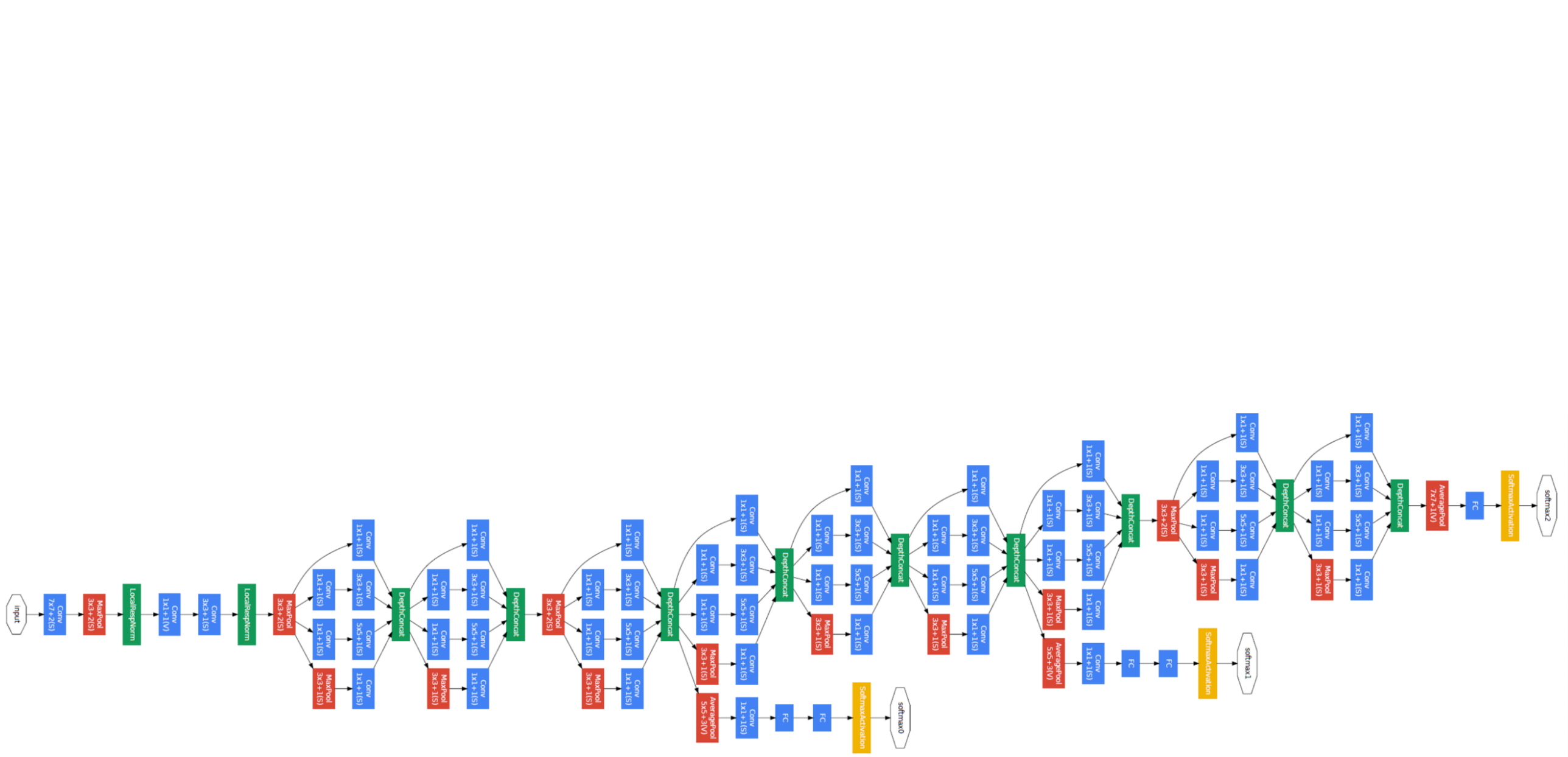
(a) Inception module, naïve version



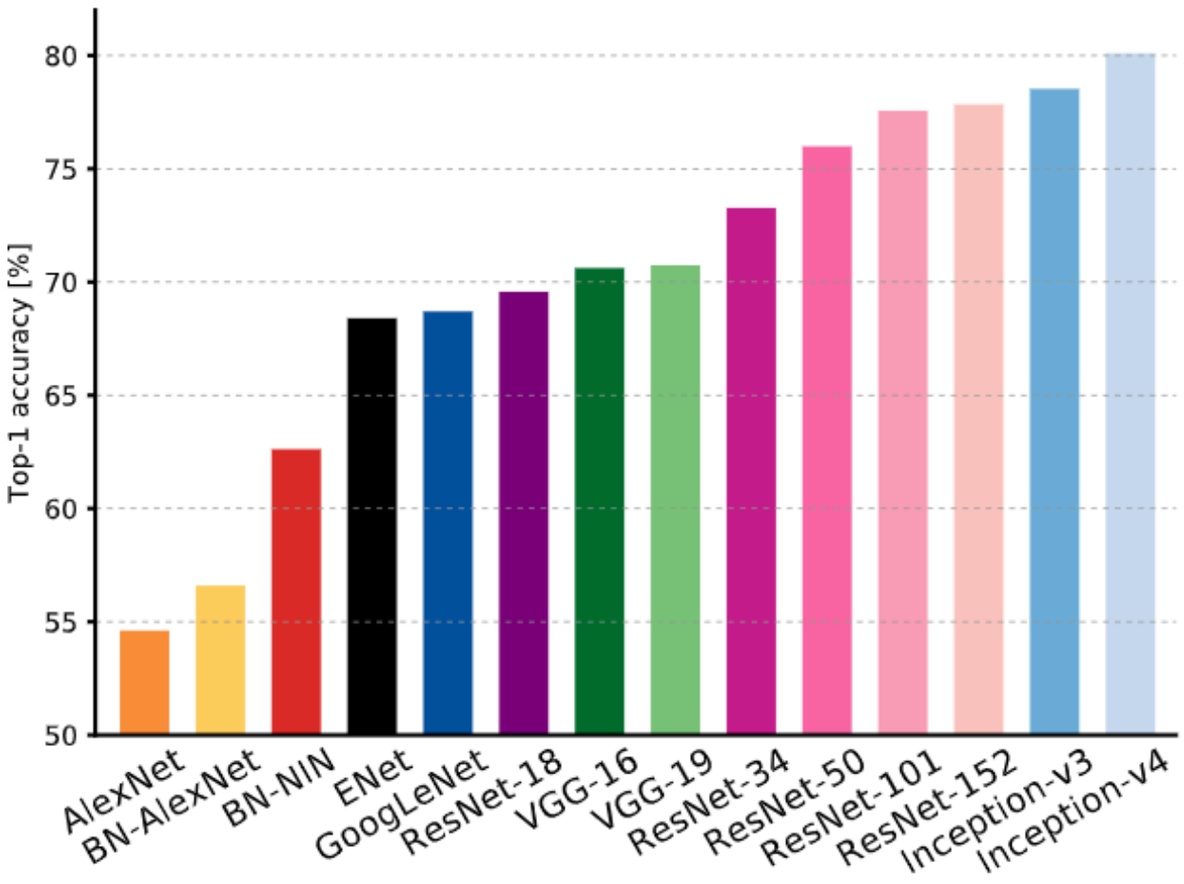
(b) Inception module with dimension reductions

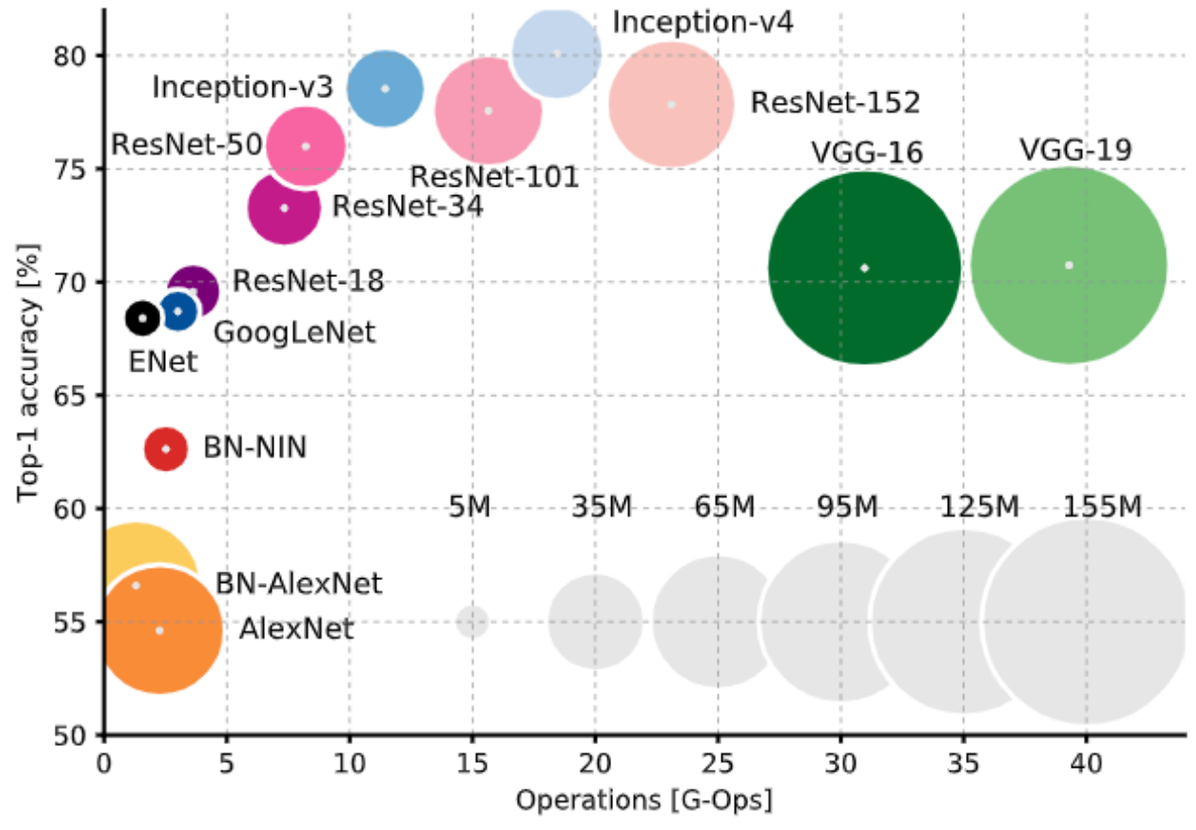
GoogLeNet

- Homage to LeNet-5 proposed by Yann LeCun et al. in 1989 (simple CNN)
- Focus on efficiency
- Introduces parallel branches of convolutions in the Inception modules (notion of going wide)
- Instead of running many networks with various convolution sizes to empirically decide on K, Inception modules learn all of them
- Naive version has much larger computational costs. 1x1 convolutions to reduce spatial channels before more expensive convolutions.
- "Stem network" at the beginning goes from 224 —> 28 spatial resolution very quickly
- Uses global average pooling and removes FC layers at end (reducing weights dramatically)
- Auxiliary classifiers for training only. The output of these is used to add to the total loss, helping combat the vanishing gradients and provides regularization.



6.7977 M parameters
9 Inception modules, 3 sofmax layers
1 global Avg pooling at end!





An Analysis of Deep Neural Network Models for Practical Applications. https://arxiv.org/pdf/1605.07678.pdf

Batch Normalisation

- Idea: to normalise the outputs of a layer so that they have zero mean and unit variance.

- This helps reduce "internal covariate shift" and improves optimisation. Weights are being updated with different data distributions across epochs. In deep networks, this results in the neurons needing to continuously adapt to the input distribution, leading to a severe bottleneck in learning capabilities.

- A batch of activations can be normalised using (which is a differentiable function)

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}}$$

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *International conference on machine learning*. PMLR, 2015.

Batch Normalisation

- The representational power of the network is compromised when each layer is normalised, as some non-linear relationships are lost. This can lead to suboptimal weights being passed on.
- To solve this issue, two learnable parameters are added

$$y = \gamma \cdot \hat{x} + \beta$$

- SGD can then tune gamma and beta to find optimal distributions. The parameters essentially scale and shift the normalised input distribution to fit the peculiarities of the given dataset.

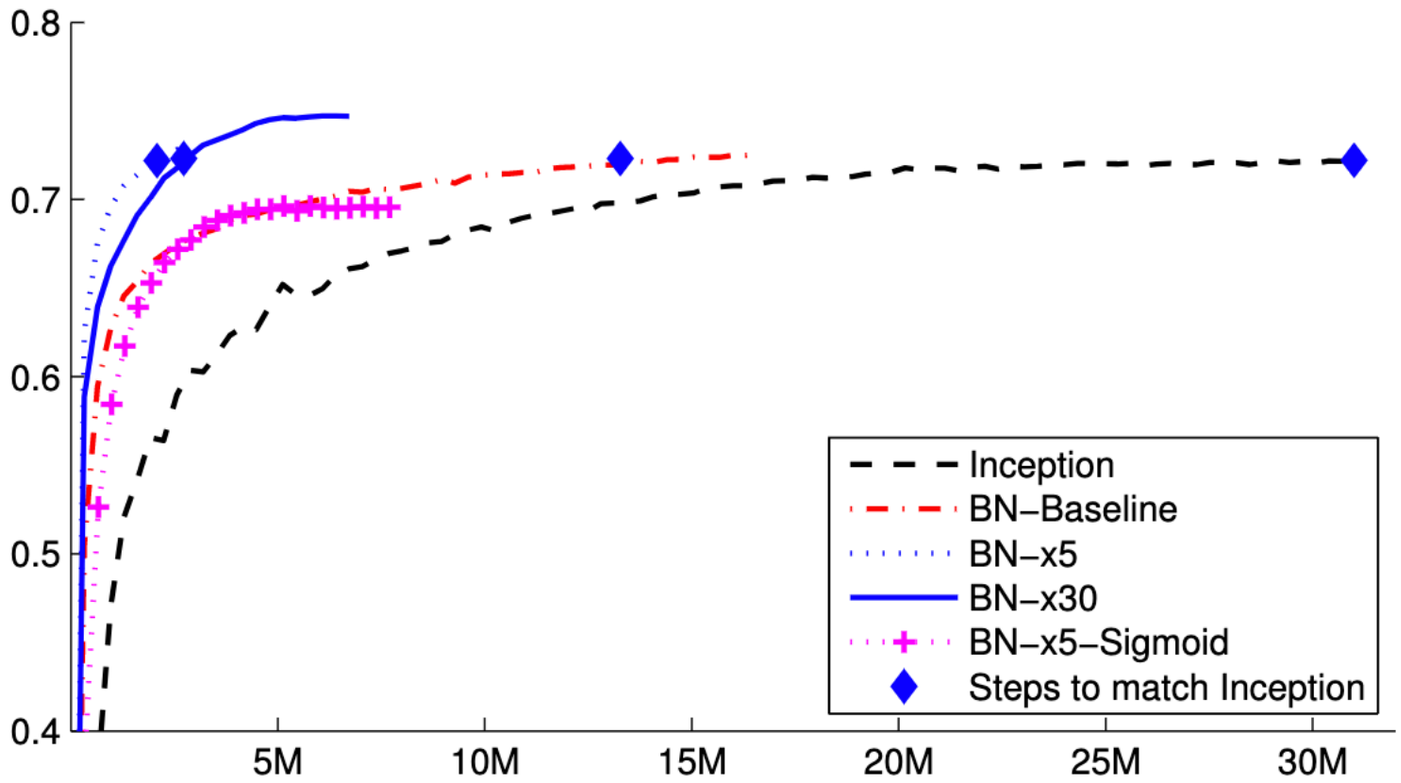
Batch Norm in Practice

- In the training phase, each batch is normalised according to the data in that particular batch.
- However, during testing, a running mean and running variance are used.

```
running_mean = momentum * running_mean + (1-momentum) * new_mean
running_var = momentum* running_var + (1-momentum) * new_var
```

- momentum term just like the one used in optimisation

Batch Normalization



- Allows training to speedup by allowing increased learning rates and Allows acceleration of learning rate decay by 6 times

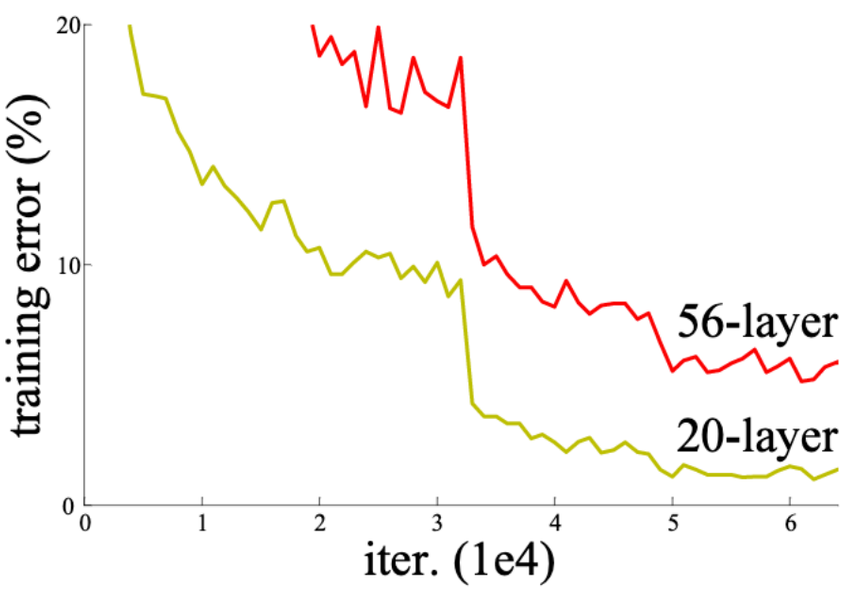
- Has a regularisation effect since it adds noise to the inputs of every layer, discouraging overfitting.

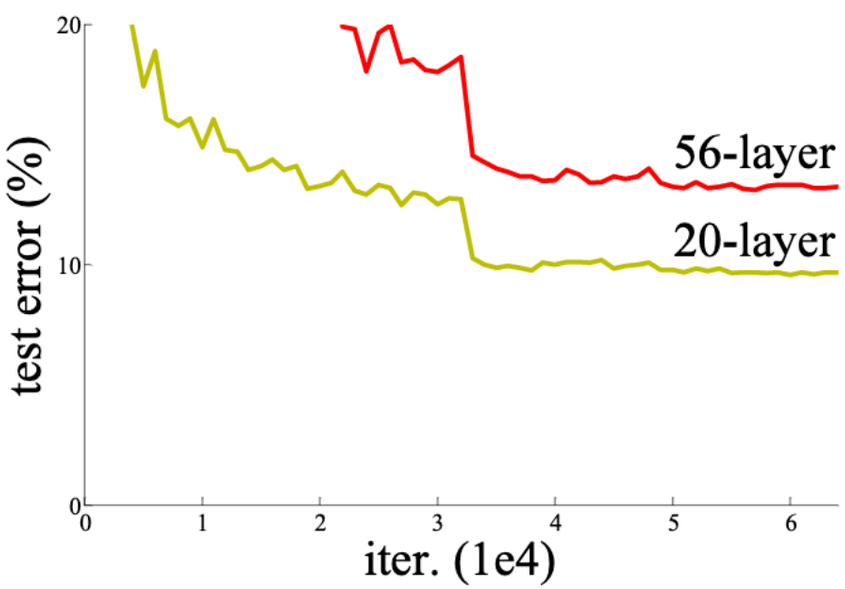
Deep Residual Networks / ResNet

Deep Residual Learning for Image Recognition. https://arxiv.org/pdf/1512.03385.pdf

Motivation

- With BatchNorm, we could now train much deeper networks
- Appears that these deeper networks are "under-fitting"
- There seems to be an optimisation problem. Deeper models are more difficult to learn





Residual Connections

- this structure will allow the network to learn the identity function

- this skip connection also helps improve the gradient flow in deep networks

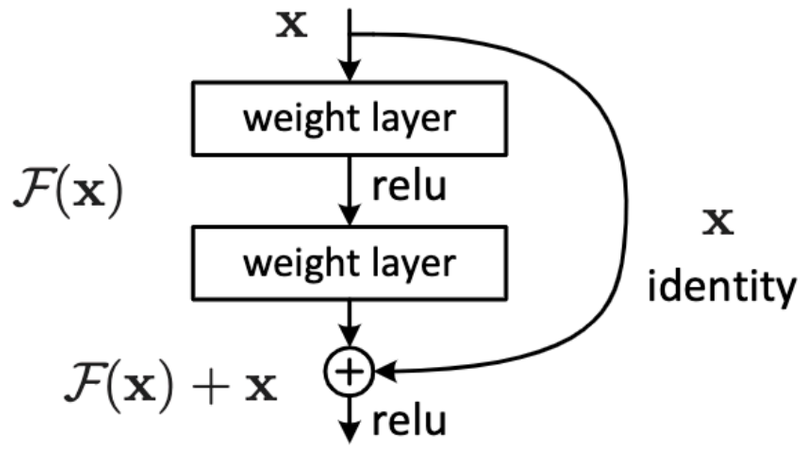
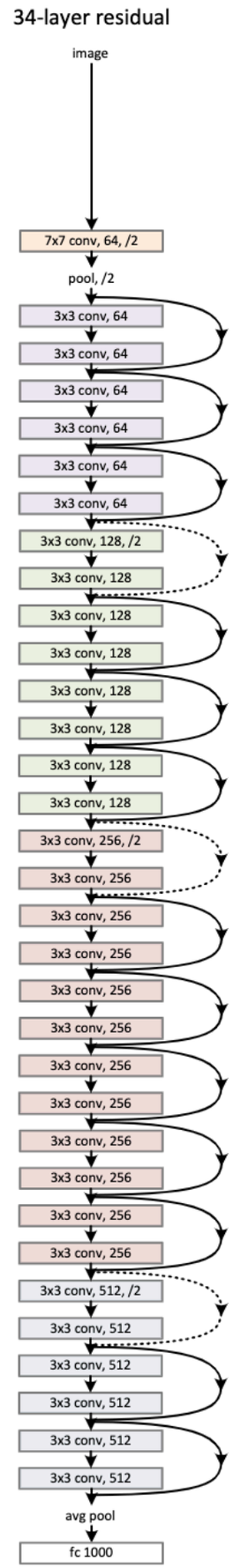


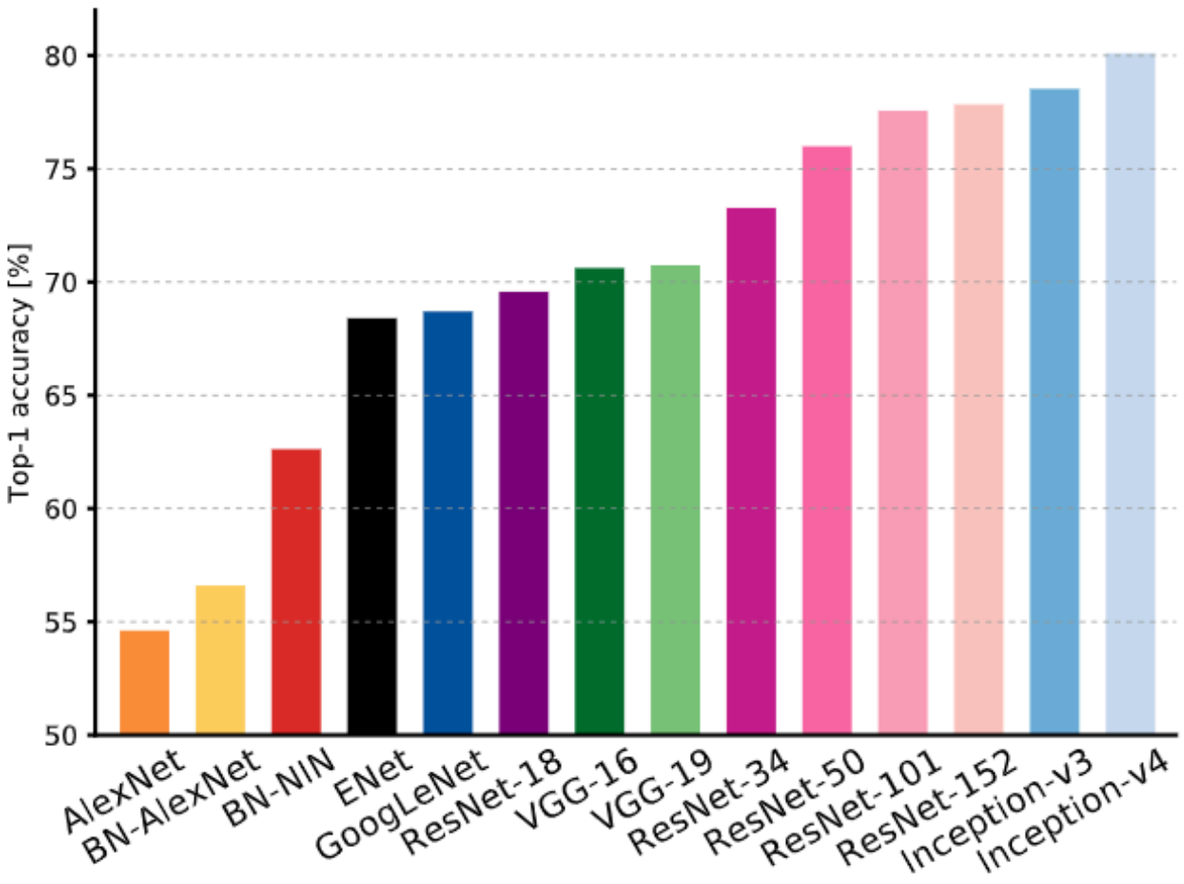
Figure 2. Residual learning: a building block.

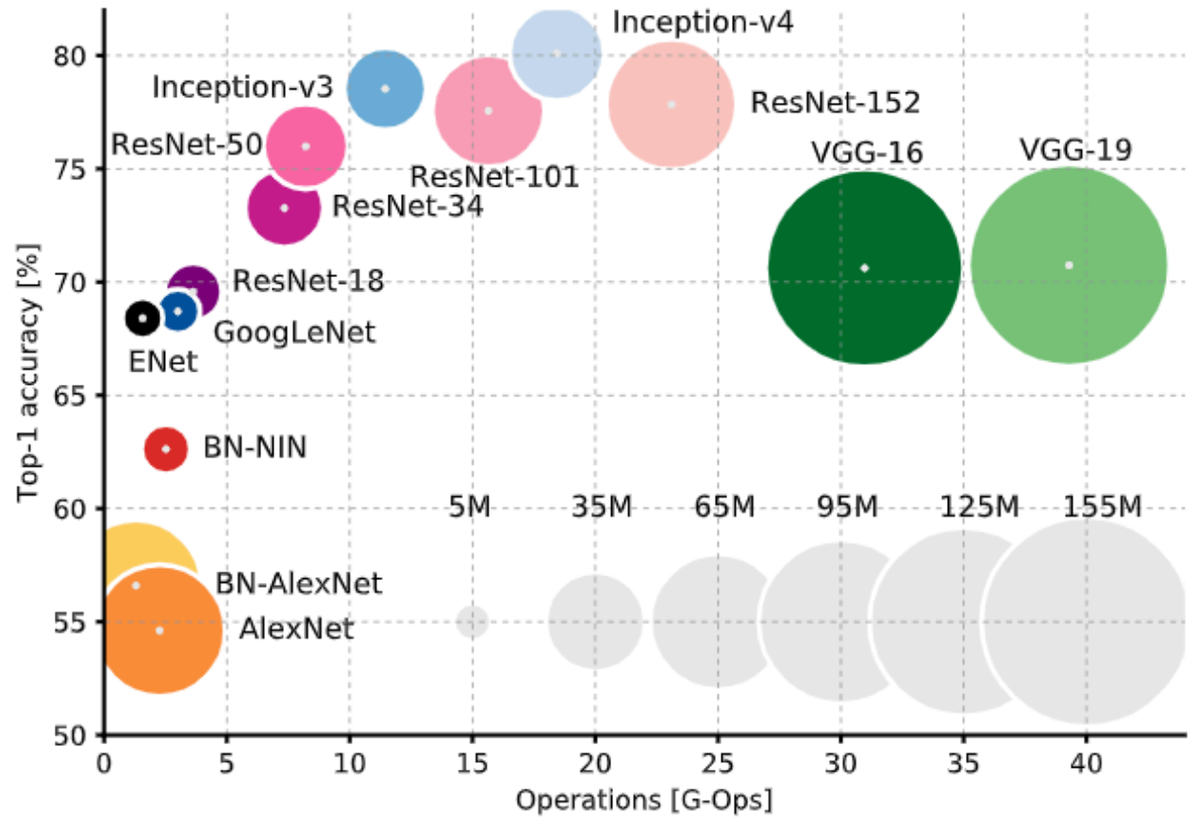
ResNets

- A ResNet is a stack of many residual blocks
- Network is divided into stages. Each stage halves the resolution (with stride 2) and double the number of channels
- Input aggressively downsamples input
- No FC layers at end, using global average pooling + single linear layer at end



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9





An Analysis of Deep Neural Network Models for Practical Applications. https://arxiv.org/pdf/1605.07678.pdf